

2. Microsoft [Электронный ресурс]. Режим доступа: <https://www.microsoft.com/ru-ru/store/b/home?irgwc=1> (дата обращения 01.04.2021 г.).

3. Троелсен Э. Язык программирования C# 5.0 и платформа .NET 4.5. 6-е изд. – М.: ООО «И.Д. Вильямс», 2015. – 1312 с.

4. Гущин А.Н. Базы данных: учебно-методическое пособие. – М., Берлин: Директ-Медиа, 2015. – 311 с.

5. ГОСТ 24104–85 ЕСС АСУ «Автоматизированные системы управления. Общие требования».

6. ГОСТ 24104–86 ЕСС АСУ «Автоматизированные системы управления. Стадии создания».

УДК 681.518

Б. Р. Саидмуродов¹, В. В. Лавров¹, Е. В. Кислицын²

¹ФГАОУ ВО «Уральский федеральный университет имени первого Президента России Б.Н. Ельцина», г. Екатеринбург, Россия

²АО «СберТех», г. Екатеринбург, Россия

РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ СБОРА И ВИЗУАЛИЗАЦИИ ДАННЫХ ИОТ-ДАТЧИКОВ

Аннотация. В процессе работы реализованы основные этапы разработки программного обеспечения: постановка задачи, создание базы данных, создание представлений, подключение базы данных, создание интерфейса пользователя, подключение приложения к серверу производителя устройств через API, хранение в базе данных показателей, измеренных удаленными беспроводными ИОТ-датчиками.

Результатом работы является работоспособное программное приложение, которое получает данные от сервера производителя датчиковой аппаратуры, сохраняет их в базу данных и отображает данные на пользовательском интерфейсе приложения в табличном виде.

Основным потребителем приложения является конечный пользователь, который, как правило, не является специалистом в программировании, но имеет базовые знания и навыки в области компьютерных технологий.

Ключевые слова: программное обеспечение, база данных, информационная система, Internet of Things (интернет-вещей и DevExpress).

Abstract. During the research, the main stages of software development were implemented: setting a problem, creating a database, creating views, connecting a database, creating a user interface, connecting an application to a device manufacturer's server via an API, storing indicators measured by remote wireless IoT sensors.

The result of the work is a workable software application that receives data from the server of the sensor equipment manufacturer, saves it to the database and displays the data in the user interface in a tabular form.

The main consumer is the end user, who usually falls into the category of non-programmers. The end user is not an expert in programming but has basic knowledge and skills in computer technology.

Key words: software, database, information system, internet of things and DevExpress.

Современный мир имеет определённые тенденции развития, одна из которых автоматизация систем удаленного доступа и использование технологии Интернета Вещей (IoT, Internet of Things) [1, 2]. Система удаленного доступа позволяет получить доступ к датчикам и исполнительным механизмам в режиме реального времени. Для этого используют такие технологии как 3G, 2G, LPWAN, RFID Z-wave, Wi-Fi, Bluetooth, LTE, Li-Fi и др. Интернет Вещей (IoT, Internet of Things) – система объединенных компьютерных сетей и подключенных физических объектов (вещей) со встроенными датчиками и программным обеспечением (ПО) для сбора и обмена данными, возможностью удаленного контроля и управления объектом в автоматизированном режиме. Благодаря беспроводной связи и различным протоколам компоненты IoT могут взаимодействовать друг с другом и отправлять собранную информацию для ее последующего анализа людьми или искусственным интеллектом.

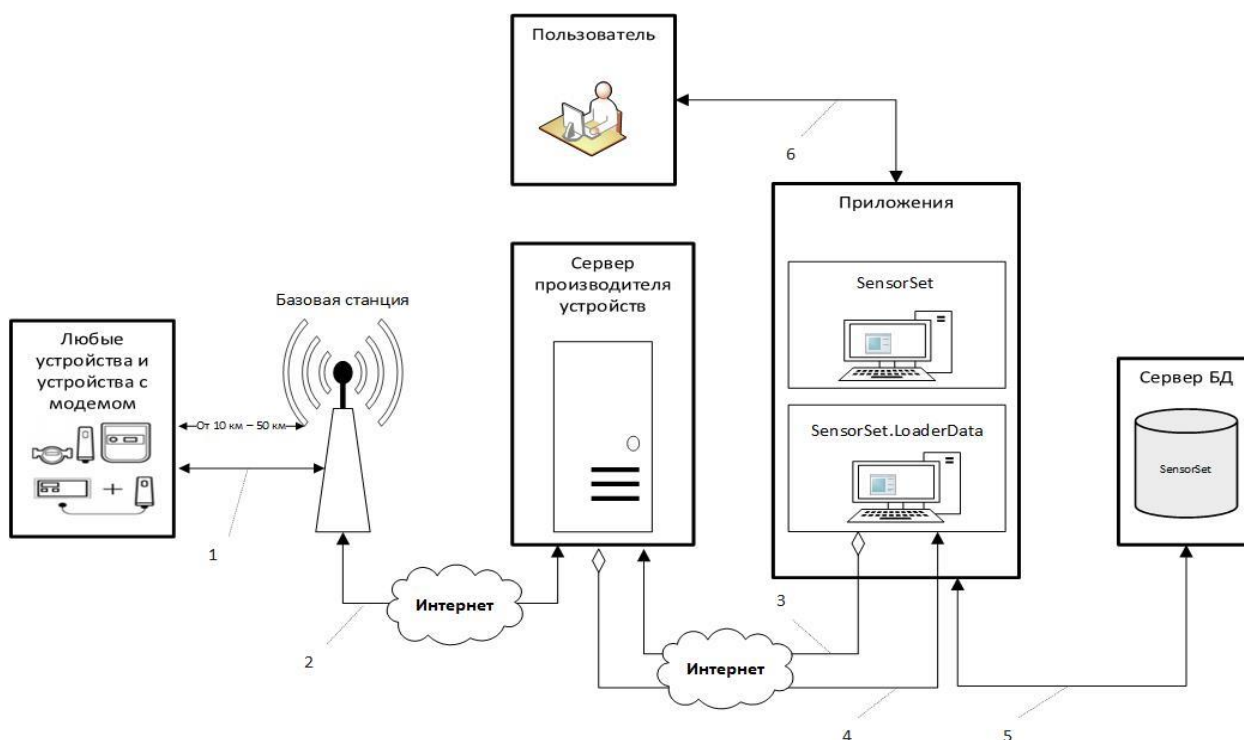
В настоящее время в Технопарке «Университетский» (г. Екатеринбург) существующее программное обеспечение системы IoT не работает номенклатурой датчиков, выпускаемых ООО «РадиоТех»: датчик качества воздуха (температуры, влажности и содержания CO₂); датчик наклона (инклинометр); датчик креномер ЛЭП; датчик протечки воды. Номенклатурная аппаратная часть не поддерживает все платформы (протоколы) для подключения к этим датчикам. В связи с этим возникла необходимость разработки собственного программного обеспечения, которое решит проблему сбора и хранения данных высокоточных измерительных датчиков в базе данных.

В данной работе представлены этапы разработки и описание функций разработанной информационной системы сбора и хранения в базе данных показателей, измеренных удаленными беспроводными IoT-датчиками.

Архитектура информационной системы представлена на рисунке 1, включает в себя следующие компоненты:

- базовая станция;
- датчики (устройства) с беспроводным доступом;
- сервер производителя устройств;
- программное приложение;
- сервер БД.

Базовая станция получает измеренные параметры IoT-датчиков и расшифровывает. Расшифрованные данные передаются на сервер производителя устройств, который их распознает по переменным (ид, значение, параметр, дата измерения) и сохраняет в БД. Пользователь с помощью приложения SensorSet.LoaderData отправляет запрос к API, в результате которого сервер передает данные, закодированные в формате json. Полученные json-файлы переводит в структуру БД и сохраняет. С помощью приложения SensorSet пользователь может посмотреть полученные данные IoT-датчиков в виде таблицы, добавить информацию, редактировать сведения об IoT-датчиках.



Информационные потоки: 1 – сбор данных из беспроводных IoT-датчиков (устройств); 2 – передача и сохранение данных в сервере производителя устройств; 3 – запрос через API в сервер производителя устройств; 4 – получение данных от сервера производителя устройств; 5 – взаимодействие БД и приложения; 6 – взаимодействие пользователя с приложением.

Рис. 1. Архитектура взаимодействия информационной системы с беспроводными удаленными IoT-датчиками

База данных реализована на платформе Microsoft SQL Server. В ней хранятся данные о различных параметрах устройств и их производителях. База данных содержит 14 таблиц: «Единицы измерения», «Измеряемые величины», «Производители», «Вид устройства», «Категория оборудования (по мобильности)», «Тип Оборудования», «Погрешность типа оборудования», «Состояние оборудования», «Параметры (значения) характеристик», «Характеристики оборудования», «Набор характеристик типового оборудования», «Экземпляры оборудования», «Погрешность оборудования» и «Измерение».

Для разработки программного приложения использована платформа Microsoft Visual Studio, язык программирования C#, библиотеки из фреймворка DevExpress. Интерфейс взаимодействия с пользователем Windows Forms. Приложение состоит из двух модулей:

1) SensorSet – служит для взаимодействия БД с пользователем, позволяет пользователю посмотреть полученные данные из IoT-датчиков, которые сохранены в БД SensorSet в виде таблицы, так же добавлять, удалять и редактировать информацию об IoT-датчиках;

2) SensorSet.LoaderData – служит для отправки пользователем запроса к API, в результате которого сервер передает данные, закодированные в формате json. Полученных json-сообщения переводить в структуру БД SensorSet и сохраняет.

При запуске программы SensorSet появляется форма регистрации, которая требует ввести имя базы данных «SensorSet» и реквизиты пользователя. После успешной регистрации пользователю отображается главная форма приложения (рис. 2).

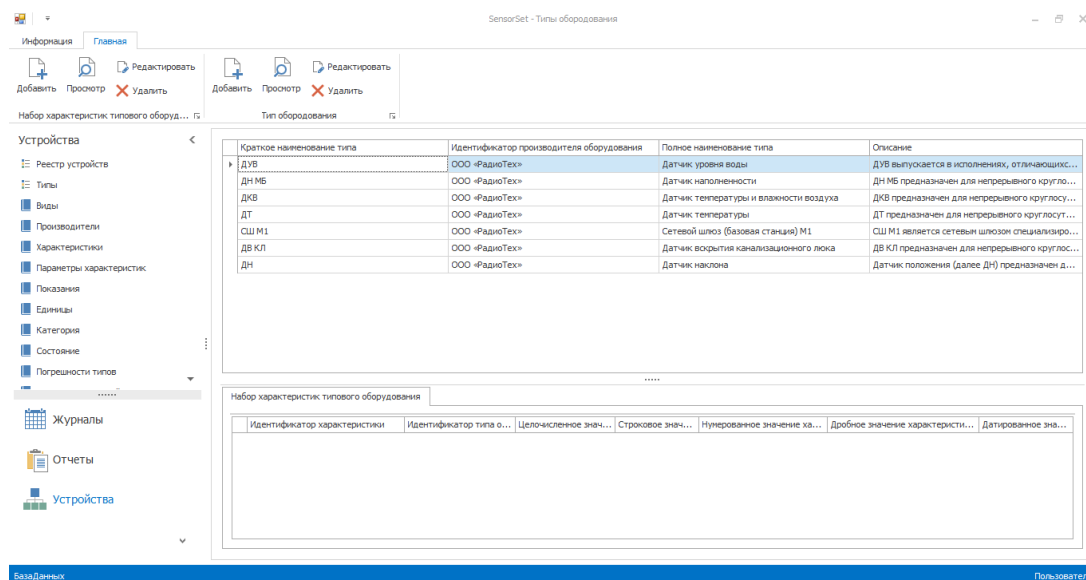


Рис. 2. Главная форма клиентского приложения SensorSet

При запуске модуля SensorSet.LoaderData появляется форма, представленная на рисунке 3.

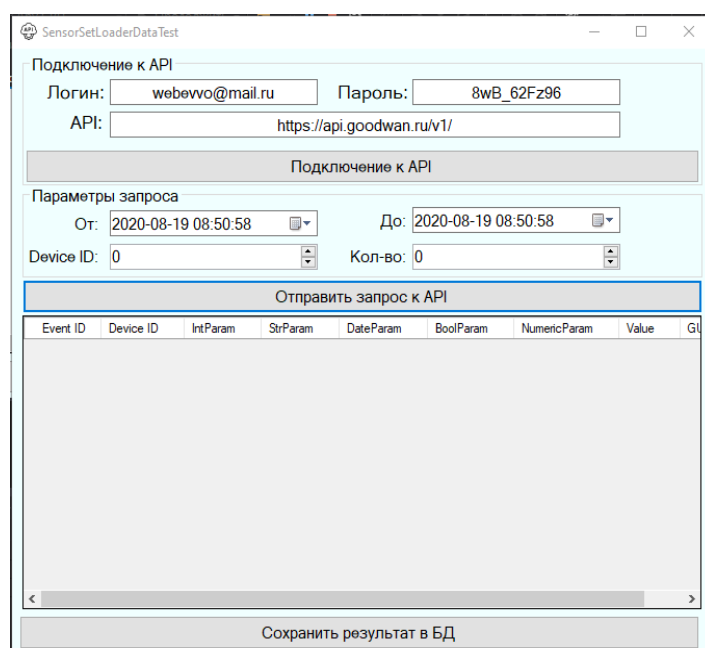


Рис. 3. Пользовательская форма модуля SensorSet.LoaderData

Пользователь в данной форме может проверить подключения к API с помощью кнопки «Подключение к API» если сервер не доступен появляется окно ошибки «Соединение не установлено» и кн. ОК, если доступен появляется окно «Соединение установлено» и кн. ОК. Задаёт параметры запроса время начало периода, конец периода, число событий и Device ID, где указывается ид устройства. При нажатии кнопки «Отправить запрос к API» на сервер производителя устройств поступает соответствующий запрос, при успешной обработке которого пользователю отображается сообщение «Парсинг завершен» и кн. ОК и полученные параметры измерения отображаются в виде таблицы. Кнопка «Сохранить результат в БД» сохраняет полученные данные в БД SensorSet.

Таким образом, в ходе выполнения работы реализованы следующие функции системы:

- спроектирован и реализован сервер БД для хранения показателей, измеренных удаленными беспроводными IoT-датчиками;
- разработано программное приложение, которое взаимодействует с сервером производителя устройств и сервером БД посредством обмена API HTTP Get запросов и обработкой json-файлов;
- создан программный модуль с удобным пользовательским интерфейсом для просмотра и фильтрации собранных показателей их БД;

Преимуществами разработанного приложения являются:

- сокращение временных затрат на сбор и обработку большого объема данных с удаленных IoT-датчиков;
- снижение нагрузки на обслуживающий персонал, исключение необходимости сбора и обработки большого количества информации на бумажных носителях;
- сокращение влияния «человеческого фактора» при мониторинге состояния объекта с возможностью оперативного устранения последствий отклонения его от нормы.

Список использованных источников

1. Зараменских Е.П. Интернет вещей. Исследования и область применения. – М.: НИЦ «Инфра-М», 2021. – 188 с.
2. Перри Л. Архитектура интернета вещей. – М.: ДМК-Пресс, 2019. – 454 с.